Motor Time Out

Local application

Thu, Jan 25, 2007

There is a need for a new LA that can monitor a motor power supply, turning off power to the motor if too much time passes since it was turned on, or if a smaller time passes since the last time the motor was adjusted. (The concern comes from the heat from the motor current putting the health of the motor itself at risk.) The LA can be called `MOTO`, for motor time out. This note describes the logic used and the design of the LA.

There is a control bit to cause the motor to be on/off. Motor adjustments can be detected by monitoring the `SETLOG` data stream for an entry that matches (listype, #bytes) and channel#, by careful monitoring of the `MOTABLE`, or by monitoring the motor countdown word in an `ADATA` table entry.

The "long" time-out parameter specifies the time after motor turn-on, in seconds, after which the LA should turn the motor off. The "short" time-out parameter specifies the time since the last motor adjustment, after which the motor should be turned off.

Consider the following LA parameter layout:

Param	*Size*	*Meaning*
`ENABLE  B`	2	Enable Bit# for LA
`MOTOR   C`	2	Motor Chan#
`LONG TO`	2	Long time-out, in seconds
`SHORT TO`	2	Short time-out, in seconds
`MOTOR   B`	2	Motor control Bit#

There is no reading of the motor position available. We could, however, monitor the motor count word to show stepping activity. If desired, we could also show the time out count down. We could also keep an internal log of the times when the LA turned off the motor. This log also might show times when the motor was turned on and thus started a long time-out. We could also include the motor adjustments made in this log, if desired.

Post-implementation notes

The implementation of `MOTO` monitors the motor countdown word in the `ADATA` table entry to notice when adjustments are made. It monitors the control line state to see when the motor is on/off, or alternatively, enabled/disabled. When it sees the motor come on, it starts the long timeout counter. While the motor is on, if it sees the motor countdown word become nonzero, it starts the short timeout counter. When the countdown word reaches zero, it restarts the short timeout counter. When the short timeout counts down to zero, the motor is turned off; *i.e.*, it is disabled. When the long timeout counts down to zero, the motor is also turned off, except that any current stepping activity is allowed to complete. Timeout logic resumes when the motor comes on again.

An internal log, meant to verify proper software functionality, has a circular buffer of entries:

Field	*Size*	*Meaning*
`logType`	2	Entry type code: '`EN`' = enable, '`DS`' = disable, '`ST`' = steps
`logSteps`	2	Current value of motor steps countdown word
`longLeft`	2	Long time-out remaining, in seconds
`shortLeft`	2	Short time-out remaining, in seconds
`logTOD`	8	Time-of-day in usual BCD format (readable in hex)